Proyecto práctico de evaluación

Tratamiento de Datos. Grado en Ciencia de Datos - UV

Carlos Santiago Martinez Torres

2025-04-26

Índice

1 Instalacion automática de paquetes

2. Asegúrate que se instalan las librerías necesarias en caso de que no estén instaladas en el ordenador en el que se ejecuta el proyecto.

```
# Asegurarse de que el paquete "pacman" está instalado
if (!require("pacman")) install.packages("pacman")
```

Una vez que el paquete pacman está instalado/cargado, establecemos las librerias que vamos a usar.

```
pacman::p_load(tidyverse, readxl, lubridate, knitr, visdat, GGally, PerformanceAnalytics)

# Con tidyverse: ggplot2, dplyr, tidyr, readr, purrr, tibble, stringr, forcats
# readxl, libreria que requiere una carga aparte explícita
# lubridate, se utiliza con frecuencia con paquetes de tidyverse, pero requiere una carga aparte
# visdat, exploracion visual de los datos. Especialmente para datos faltantes
# GGally, graficar correlaciones de variables
# PerformanceAnalytics, Crear gráfico de correlacion en R con histogramas, funciones de densidad, linea
```

2 Carga de datos

3. Carga el fichero de datos en R y llama datos al data frame obtenido. Este data frame contendrá exclusivamente las columnas con variables y las filas que contengan datos. Se sabe que los datos empiezan en la fila 25 pero NO cuántos alumnos puede haber por lo que la determinación selección de filas con datos debe ser AUTOMÁTICA.

```r
# nombre_fichero <- 'NosAutoanalizamos2022 - Hoja1.tsv'
nombre_fichero <- 'NosAutoanalizamos2025 - Hoja1.tsv'
fichero <- file.path('./data', nombre_fichero)

datos <- read.delim(file=fichero, header=FALSE, sep='\t')
```

3 Limpieza de datos

Una vez que ya cargamos el fichero, vamos a quedarnos solo con los valores que nos interesan.

```r
# Informacion sobre los datos almacenados, fijo
InfoCode <- datos[1:19, ]

# Sabemos que los datos empiezan a partir de la linea 25
# Filtramos desde las primeras 25 lineas
datos <- datos[25:nrow(datos), ]
```

Buscamos el registro que contiene 'FechaNac', ya que a partir de ahí, tomaremos los datos a trabajar.

```r
# Al conjunto datos, por cada fila(1) -2 para columma-, evaluo si cualquier valor es 'FechaNac'
f1 <- which(apply(X = datos, MARGIN = 1, FUN = function(fila) any(grepl('FechaNac', fila))))[[1]]
f2 <- which(apply(X = datos, MARGIN = 2, FUN = function(col) any(grepl('FechaNac', col)))) # Para la co
col_FechaNac <- names(datos)[f2]

# Filtrar desde f1 hasta el final, con todas las filas
datos <- datos[f1:nrow(datos), f2:ncol(datos)]
```

Ahora tomamos las columnas que contienen las variables de interes.

```r
# Seleccionamos las variables de interes en la informacion de las variables
f3 <- which(apply(X = InfoCode, MARGIN = 2, FUN = function(col) any(grepl('Fecha', col))))
col_info <- names(InfoCode)[f3]

InfoCode <- InfoCode %>%
              select(all_of(col_info))

colnames(InfoCode) <- c('Variable', 'Descripcion')
```

Dado que los pares del conjunto, fecha y id no se debería repetir, eliminamos aquellos datos que estan parcialmente duplicados. Esto ayuda a eliminar los registros vacios.

```r
# # No empleamos estas funciones, ya que no hay valores NA, y de haberlo un regitro con al menos un NA,

# # Eliminar filas que tengan todos los valores vacíos (NA)
# datos[!apply(is.na(datos), 1, all), ]
#
# # Eliminar filas que tengan algún valor vacío (NA)
# datos[complete.cases(datos), ]

f4 <- which(apply(X = datos, MARGIN = 2, FUN = function(fila) any(grepl('Id', fila)))) # Para la col qu
```

Cuadro 1: Informacion de los datos almacenados

Variable	Descripcion
Fecha	Fecha de nacimiento en formato 29-01-2001
Identificación	Usuario de correo UV. martsobm
Age	Edad en años.
Sex	Sexo de nacimiento: F/M
Wr.Hnd	Distancia del extremo del meñique al extremo del pulgar de la mano con la que escribimos. (en centímetro
NW.Hnd	Mano con la que escribes: Izquierda, Derecha
Fold	Cruza los brazos. Qué brazo está arriba ?: Izquierdo, Derecho
Pulse	Cuál es tu ritmo cardíaco, en reposo, en latidos por minuto.
Clap	Aplaude. ¿Cómo están las manos ?. 3 casos. 'Der s Izd', (Derecha sobre izquierda) ' Izq s Der', 'Ninguna'
Smoke	Fumas: Si / No
Exer	Cuántos días a la semana realizas ejercicio físico. Numérica 0-7
Height	Altura (en centímetros)
ALG	Calificación numérica obtenida en álgebra. Usa NA para no presentado y NC, para no compensable, si pro
ANM	Calificación numérica obtenida en análisis matemático. Usa NA para no presentado y NC, para no compen
FP	Calificación numérica obtenida en fundamentos de programación. Usa NA para no presentado y NC, para
DCS	Calificación numérica obtenida en Datos Ciencia y Sociedad. Usa NA para no presentado y NC, para no c
MD	Calificación numérica obtenida en Matemática Discreta. Usa NA para no presentado y NC, para no compe
HSt	Horas de trabajo DIARIAS relacionadas con los estudios, además de la asistencia a clase.
Hwork	¿Tienes un trabajo remunerado además del estudio ?. Indica las horas de trabajo SEMANALES. 0 indica

```
col_Id <- names(datos)[f4]

datos <- datos %>%
            distinct(across(all_of(c(col_FechaNac, col_Id))), .keep_all = TRUE)
```

Si comprobamos el df, vemos que aún hay registros que no son relevantes para nuestro analisis.

```
# La primer fila del df debe ser el nombre de las columnas
head(datos, 3)

# Hay un registro con todos los campos vacios
tail(datos, 3)
```

Extraemos el nombre de las columnas, que están en el primer registro del df, para asignarlos correctamente.

```
# Obtenemos el vector de los nombres de las columnas
nombres <- as.character(datos[1, ])

# Del df de datos, elimamos el registro que contiene los nombres
datos <- datos[-1, ]
```

```
kable(InfoCode,
      caption= 'Informacion de los datos almacenados',
      booktabs= TRUE)
```

Siguiendo el formato del CodeBook que se observa en la Tabla 1, es como vamos a actualizar los nombres del df.

Cuadro 2: Registros almacenados

Fecha	Id	Age	Sex	Wr.Hnd	NW.Hnd	Fold	Pulse	Clap	Exer	Smoke	Hei
20-01-1969	martsobm	55	M	22	Derecha	Izquierdo	65	Der s Izq	2	No	178
22-02-2006	olrae	18	Masculino	20	dercha	izquierdo	60	Der s Izq	12	No	173
07-11-2006	jusieal	18	M	20	Derecha	Izquierdo	62	Der s Izq	1	De fiesta	180
28-01-2004	jovimo3	21	Masculino	18	Derecha	Izquierdo	65	Der s Izq	3	No	173
11-10-2003	carmart4	21	M	20	Derecha	Izquierdo	69	Der s Izq	3	No	172

```
# Actualizamos los nombres necesarios de acuerdo al CodeBook
nombres[1] <- 'Fecha'

# Actualizamos los nombres en el df
colnames(datos) <- nombres
```

Antes de continuar, quitaremos los registros vacios. Para ello, sabiendo que tanto la fecha como el id son valores no vacios, filtramos por aquellos que si lo son.

```
datos <- datos %>%
  filter(!Fecha== '' | !Id== '')
```

Podemos comprobar los datos almacenamos en el df datos, que se muestran en la tabla 2.

Verificamos el tipo de dato que tenemos en cada una de las variables de nuestro dataframe.

```
glimpse(datos)
```

```
## Rows: 59
## Columns: 20
## $ Fecha       <chr> "20-01-1969", "22-02-2006", "07-11-2006", "28-01-2004", "1~
## $ Id          <chr> "martsobm", "olrae", "jusieal", "jovimo3", "carmart4", "an~
## $ Age         <chr> "55", "18", "18", "21", "21", "18", "19", "18", "18", "18"~
## $ Sex         <chr> "M", "Masculino", "M", "Masculino", "M", "M", "M", "M", "F~
## $ Wr.Hnd      <chr> "22", "20", "20", "18", "20", "21", "16", "17", "9,5", "16~
## $ NW.Hnd      <chr> "Derecha", "dercha", "Derecha", "Derecha", "Derecha ", "De~
## $ Fold        <chr> "Izquierdo", "izquierdo", "Izquierdo", "Izquierdo", "Izqui~
## $ Pulse       <chr> "65", "60", "62", "65", "69", "65", "92", "62", "65", "70"~
## $ Clap        <chr> "Der s Izq", "Der s Izq", "Der s Izq", "Der s Izq", "Der s~
## $ Exer        <chr> "2", "12", "1", "3", "3", "8", "si", "", "3", "2", "4", "2~
## $ Smoke       <chr> "No", "No", "De fiesta", "No", "No", "Si", "no", "No", "No~
## $ Height      <chr> "178", "173", "180", "173", "172", "179", "185", "170", "1~
## $ ALG         <chr> "NP", "7,2", "6", "6,2", "8,5", "3,3", "6,8", "3,5", "NC",~
## $ ANM         <chr> "7", "6,2", "7", "6,6", "8,2", "6,3", "7,8", "3,7", "7,2",~
## $ FP          <chr> "6", "NP", "3,91", "9", "9", "0,65", "2,75", "1,8", "3", "~
## $ DCS         <chr> "4", "4", "7,6", "8,3", "9", "6,4", NA, "4", "7,9", "8,6",~
## $ MD          <chr> "3,8", "1,7", "5,3", "5", "8", "5", "6,3", "2,5", "6,2", "~
## $ HSt         <chr> "5", "1,5", "1", "3", "6", "2", "0", "0,5", "2", "4", "", ~
## $ Hwork       <chr> "37,5", "5", "0", "0", "25", "12", "0", "0", "0", "2", "",~
## $ Comentarios <chr> "Seguro que este grupo es el mejor que he tenido hasta aho~
```

En la siguiente figura 1 se identifican los valores NA. Habrá que hacer la conversión al tipo correcto de dato para ver que inconsistencias podemos tener.

```
vis_miss(datos)
```

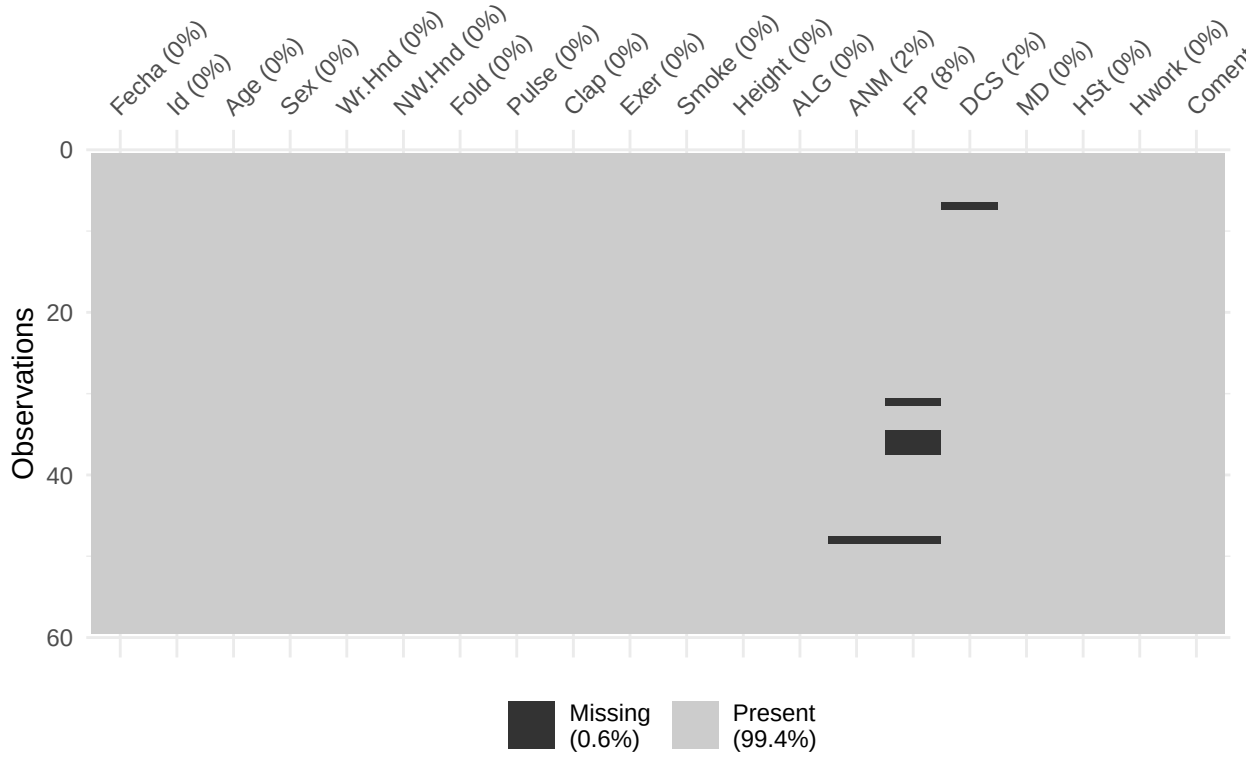


Figura 1: Visualizacion de datos NA

3.1 Tratamiento de datos

4. Elimina al usuario cuyo id es martsobm del conjunto de datos.

```
datos <- datos %>%
  filter(!Id =='martsobm')
```

5. Selecciona las columnas con las calificaciones y adecúalas para que al transformar a tipo numérico los valores sean correctos (recuerda que el separador decimal es el punto). Si alguna calificación es NC se sustituirá por 3. Cualquier otro valor no conocido se sustituirá por NA (por ejemplo "6 y pico", no es un valor conocido).

Seleccionamos las columnas de las calificaciones

```
inicio_cal <- which(colnames(datos) == 'ALG')
final_cal <- which(colnames(datos) == 'MD')
```

```r
# across: permite aplicar una misma operacion a varias columnas de un mutate() o summarise()
# .cols: seleccion de columnas
# everything(), starts_with(), ends_with(), matches(), contains(), c(col1, col2), col1:col5
# .fns: funcion o lista de funciones a aplicar a cada columna seleccionada.
# .names: El nombre de las nuevas columnas. Por defecto, FALSE.

# Vemos que no hay valores o expresiones como '6 y pico'. Por lo tanto podremos aplicar as.numeric al d

# Si solo se está filtrando, no es necesario hacer un select.

datos %>%
  filter(if_any(.cols = inicio_cal:final_cal,
                .fns = ~ str_length(.) >= 4))
```

Aplicamos, para cada columna, la condicion, haciendo el respectivo cambio de tipo de dato.

```r
datos[inicio_cal:final_cal] <- datos %>%
                               select(inicio_cal:final_cal) %>%

                               # Reemplazamos para que el separador decimal sea el punto
                               mutate(across( .cols = everything(),
                                              .fns = ~ str_replace_all(., ',', '.'))) %>%

                               # Personalizamos la funcion, para pasar una lista a ifelse
                               mutate(across( .cols = everything(),
                                              .fns = ~ifelse(.=='NC', 3, round(as.numeric(.), 1)))) # 
```

6. Obtén una tabla de estadísticos de estas variables. Donde en la primera columnas aparezca cada una de las asignaturas y las siguientes columnas se correspondan con los estadísticos que se indican, en este orden: valor mínimo, percentil25, mediana, valor medio, desviació típica, percentil75, valor máximo. Los resultados se mostrarán con 2 decimales.

```r
df_notas <- datos[inicio_cal:final_cal]
df_estadisticos_notas <- df_notas %>%
  pivot_longer(cols = ALG:MD,
               names_to = 'Asignatura',
               values_to = 'Nota') %>%
  group_by(Asignatura) %>%
  summarise(across(.cols = everything(),
                   .fns = list(minimo = ~ min(., na.rm = TRUE),
                               percentil25 = ~ quantile(., probs = 0.25, na.rm = TRUE),
                               mediana = ~ median(., na.rm = TRUE),
                               medio = ~ mean(., na.rm = TRUE),
                               desviaciontipica = ~ sd(., na.rm = TRUE),
                               percentil75 = ~ quantile(., probs = 0.75, na.rm = TRUE),
                               maximo = ~ max(., na.rm = TRUE)))) %>%
  mutate(across(.cols = starts_with('Nota'),
         .fns = ~ round(., digits = 2)))

kable(df_estadisticos_notas,
      caption = 'Tabla de estadisticos sobre notas',
      booktabs = FALSE)
```

Cuadro 3: Tabla de estadisticos sobre notas

Asignatura	Nota_minimo	Nota_percentil25	Nota_mediana	Nota_medio	Nota_desviaciontipica	Nota_percen
ALG	2.2	3.3	6.0	5.32	1.89	
ANM	3.0	6.0	7.0	6.78	1.52	
DCS	3.0	6.0	7.0	6.75	1.53	
FP	0.7	3.0	5.0	4.96	2.13	
MD	1.7	5.0	5.5	5.43	1.58	

Ver tabla 3.

7. Usa la función ggpairs de la librería GGally y representa las relaciones entre todas las variables.

```
# Cargada posteriormente con ggplot2
# library(GGally)
df_estadisticos_notas %>%
  select(-Asignatura) %>%
  ggpairs(.)
```

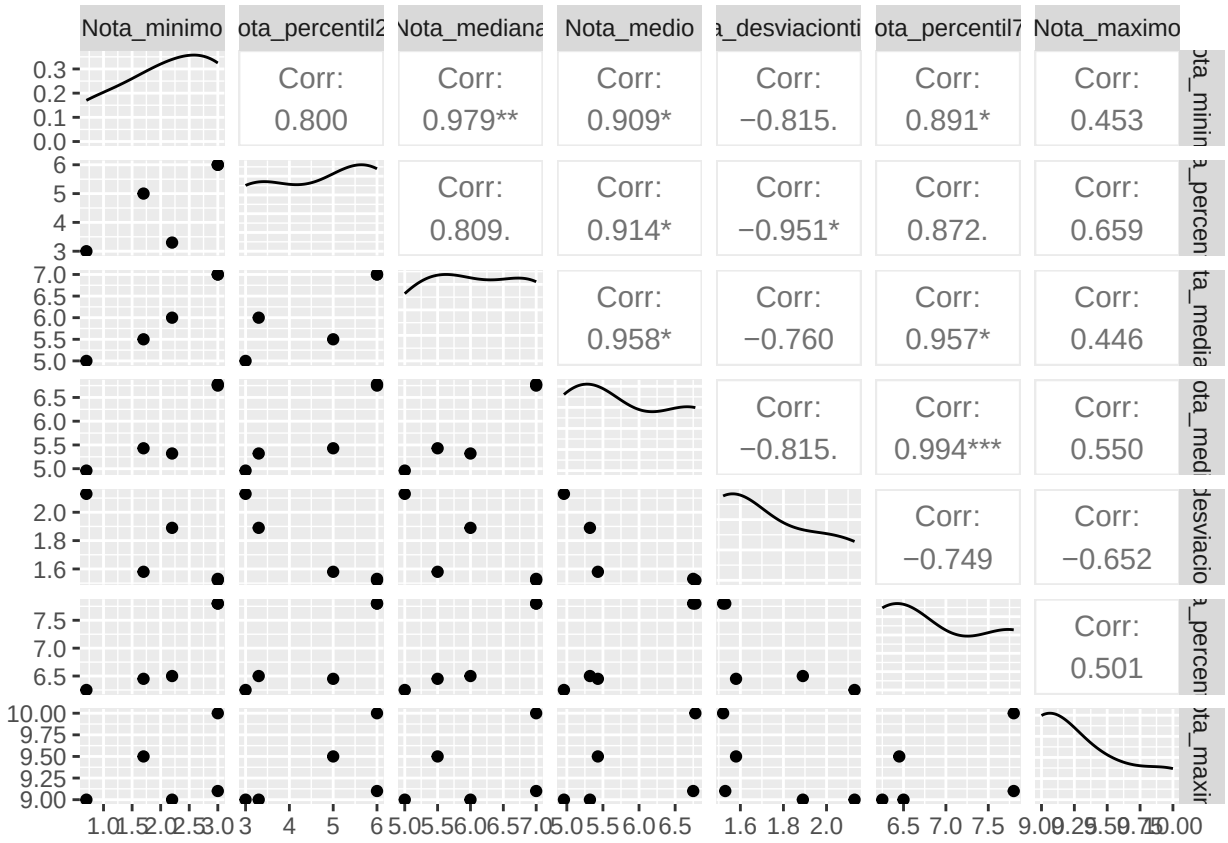


Figura 2: Relacion entre todas las variables

Podemos ver en la figura 2 que hay una relación positiva entre la nota media, mediana y el percentil 75 que obtuvieron los estudiantes de tal curso.

8. Determina las matrices de correlación y covarianza de dichas variables, con las opciones “Spearman” y “Pearson” e indica qué pares de variables tienen la correlación más alta.

```
# Las estrellas permiten saber si la variable es estadisticamente significativa.

# install.packages("PerformanceAnalytics")
# library(PerformanceAnalytics)
```

```
# # Correlacion
# df_estadisticos_notas %>%
#   select(-Asignatura) %>%
#   cor(., method = 'pearson')

# # Covarianza
# df_estadisticos_notas %>%
#   select(-Asignatura) %>%
#   cov(., method = 'pearson')

df_estadisticos_notas %>%
  select(-Asignatura) %>%
  chart.Correlation(., histogram = TRUE, method = 'pearson')
```

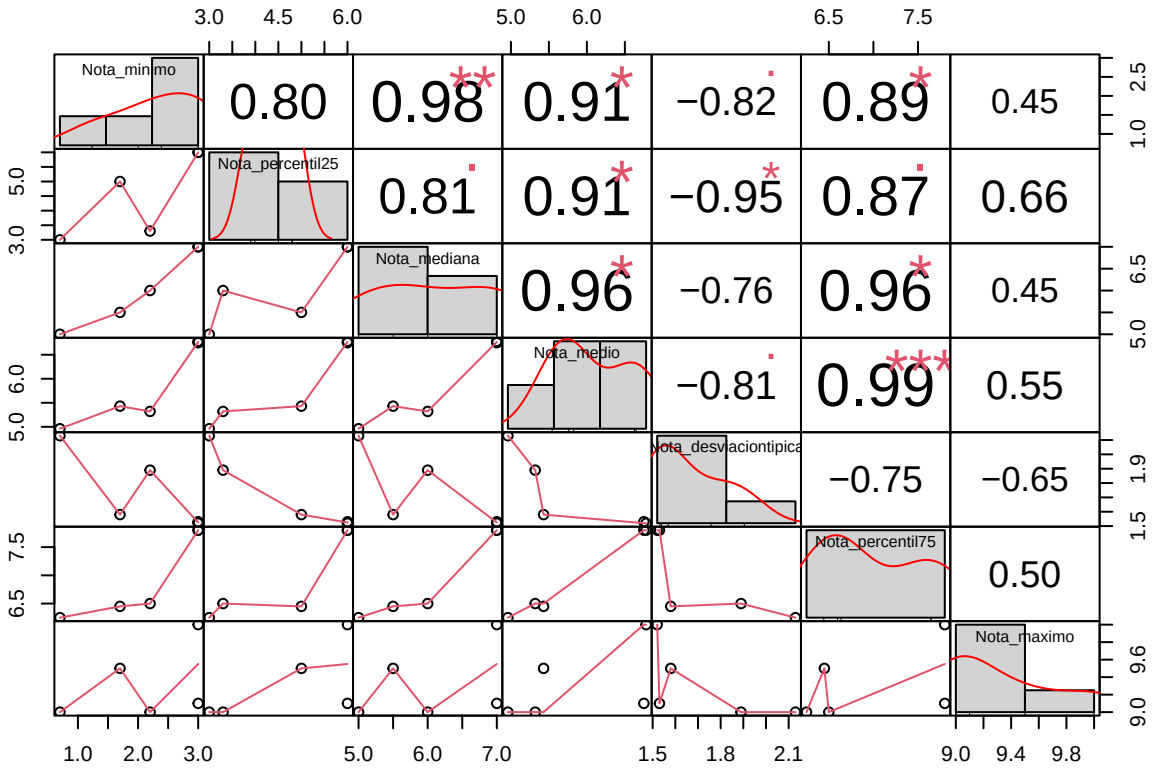


Figura 3: Matriz de correlación de Pearson

```r
# # Correlacion
# df_estadisticos_notas %>%
#   select(-Asignatura) %>%
#   cor(., method = 'spearman')

# # Covarianza
# df_estadisticos_notas %>%
#   select(-Asignatura) %>%
#   cov(., method = 'spearman')

df_estadisticos_notas %>%
  select(-Asignatura) %>%
  chart.Correlation(., histogram = TRUE, method = 'spearman')
```

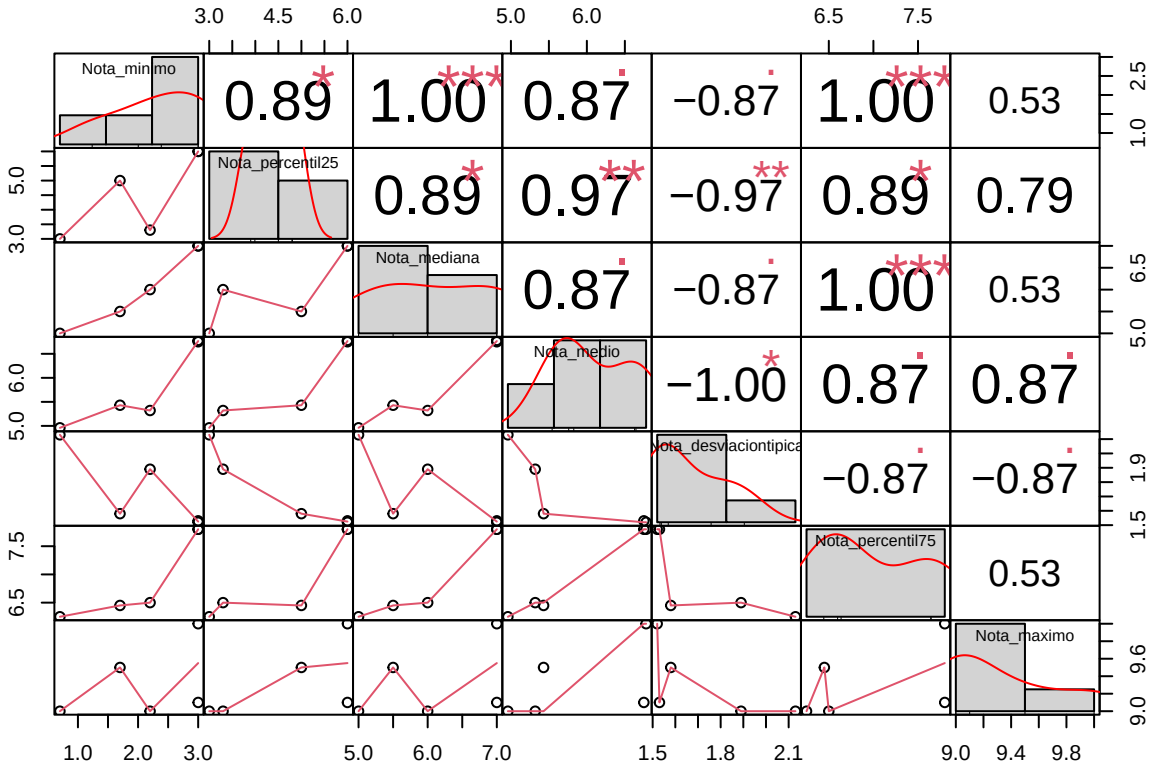


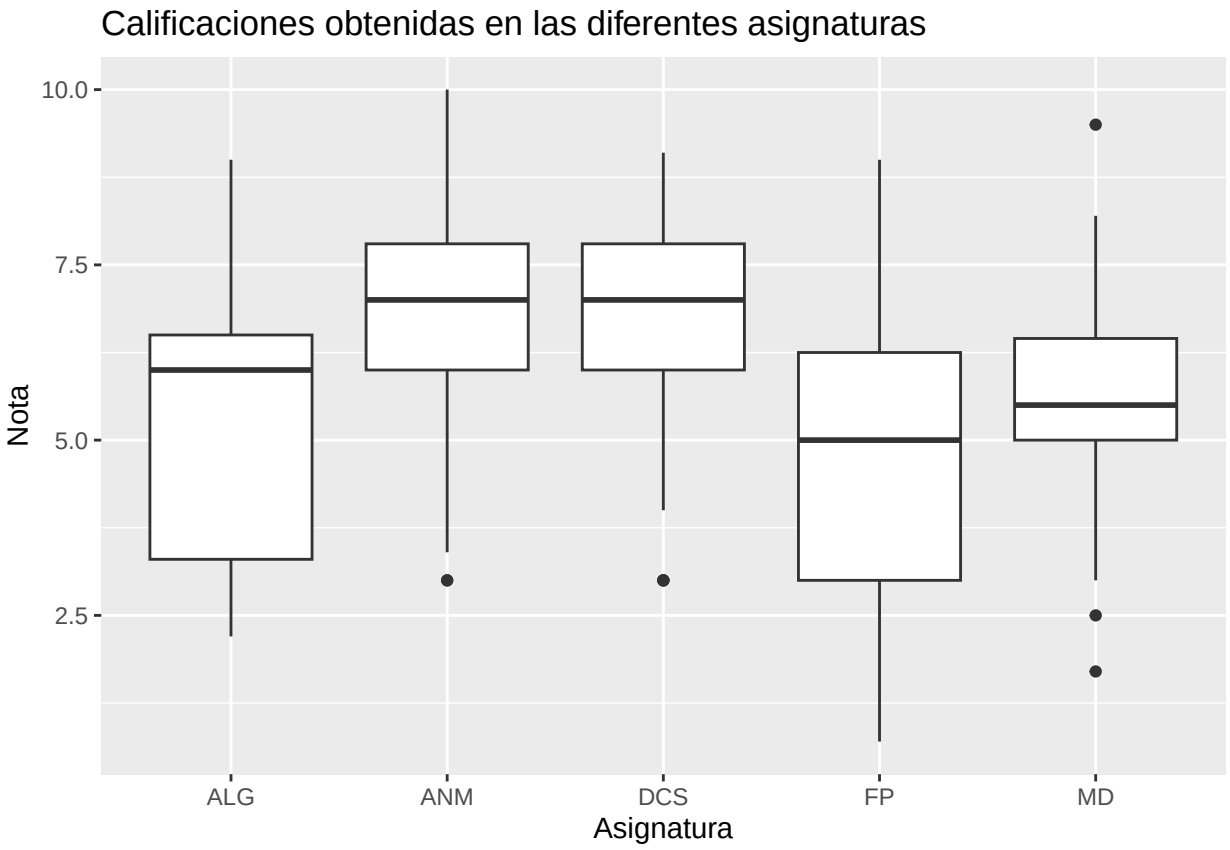
Figura 4: Matriz de correlación de Spearman

En las figuras 3 y 4 se puede identificar que las variables con la correlación más alta son percentil 75, media y mediana.

9. Representa en una misma gráfica los boxplots con las calificaciones obtenidas en las diferentes asignaturas.

```r
df_notas %>%
  pivot_longer(cols = ALG:MD,
```

```
              names_to = 'Asignatura',
              values_to = 'Nota') %>%
  ggplot(aes(x=Asignatura, y=Nota)) +
  geom_boxplot(na.rm=TRUE) +
  labs (title = 'Calificaciones obtenidas en las diferentes asignaturas')
```



10. Añade la variable sexo al conjunto de datos y realiza la representación del boxplot de calificaciones por asignatura coloreando según dicha variable.

Correccion sobre la columa sexo.

```
# Corregir los valores de la columna Sex
datos <- datos %>%

        # Quitar los espacios en blanco
            mutate(Sex = str_trim(Sex)) %>%

        # Dejarlas en mayuscula
            mutate(Sex = str_to_upper(Sex))

# Normalizar los valores
datos %>%
  count(Sex)
```

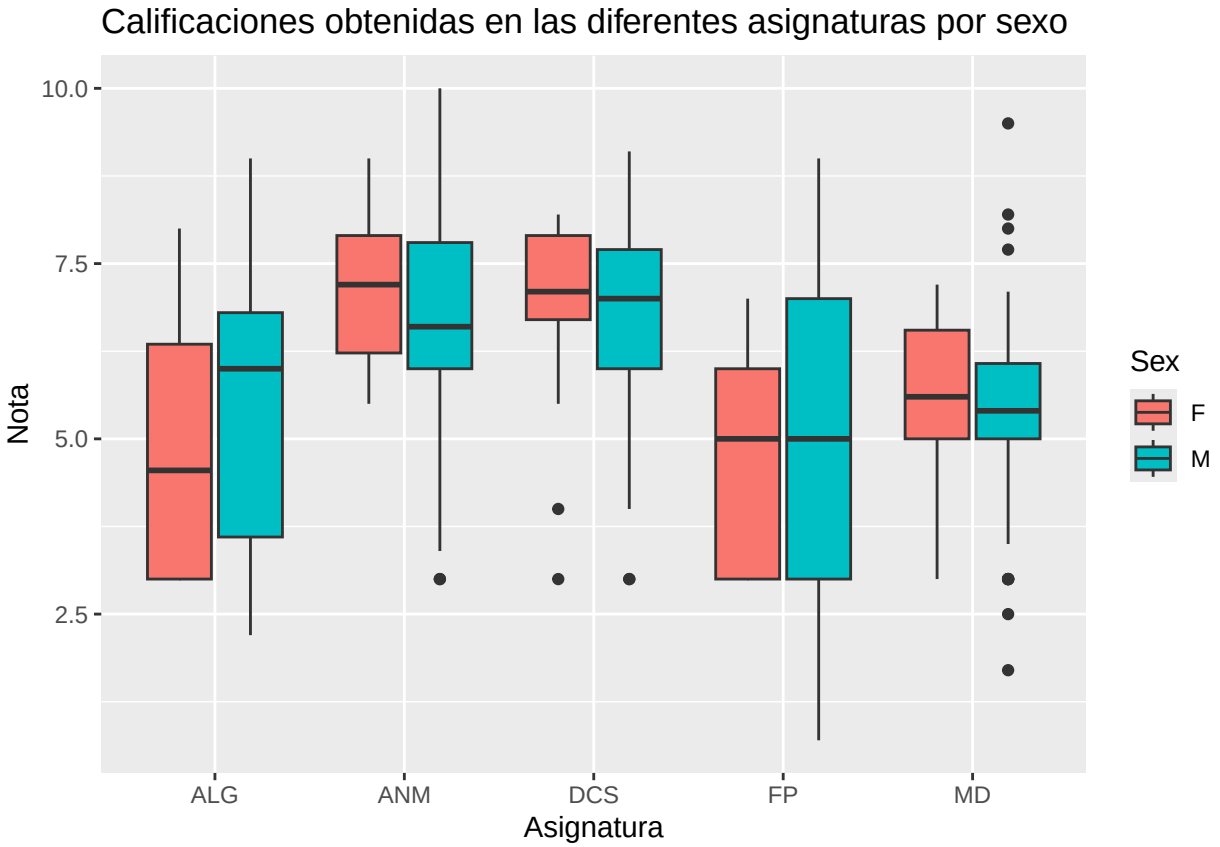
Sex	n
F	17
M	37
MASCULINO	4

```
# Guardamos los valores en un vector
inputs <- datos %>%
  distinct(Sex) %>%
  as.vector()
inputs <- inputs[[1]]

# Identificamos los que pueden referirse a M y F
femenino <- inputs[which(str_detect(string = inputs, pattern = 'F'))]
masculino <- inputs[which(str_detect(string = inputs, pattern = 'M'))]

# Actualizamos el dataframe, colapsando los valores
datos <- datos %>%
            mutate(Sex = fct_collapse(Sex, M = masculino)) %>%
            mutate(Sex = fct_collapse(Sex, F = femenino))
```

```
datos %>%
  select(Sex, ALG:MD) %>%
  pivot_longer(cols = ALG:MD,
               names_to = 'Asignatura',
               values_to = 'Nota') %>%
  ggplot(aes(x=Asignatura, y=Nota, fill=Sex)) +
  geom_boxplot(na.rm=TRUE) +
  labs (title = 'Calificaciones obtenidas en las diferentes asignaturas por sexo')
```



11. Selecciona los alumnos que tiene calificaciones en todas las asignaturas y represéntalas gráficamente. En el eje X aparecerá el usuario del alumno (girado 90º) y en el eje Y la calificación numérica con un color distinto para cada asignatura. Se mostrará un diagrama de líneas, que una las calificaciones de una misma asignatura para todos los alumnos así como un punto, por asignatura (del mismo color que las líneas), para cada asignatura y alumno.

```
# CASOS COMPLETOS
datos %>%
  select(Id, ALG:MD) %>%
  #complete.cases()
  na.omit() %>%

  # CONVERSION A TIDY
  pivot_longer(cols = ALG:MD,
               names_to = 'Asignatura',
               values_to = 'Nota') %>%
  group_by(Id) %>%
  #mutate(Id = factor(Id)) %>%

  # GRAFICO
  ggplot(aes(x=Id, y=Nota, group=Asignatura)) +
  geom_line(aes(color = Asignatura)) +
  #geom_line(aes(linetype = Asignatura, color = Asignatura)) +
  geom_point(aes(color = Asignatura)) +
  theme(axis.text.x = element_text(angle = 90))
```

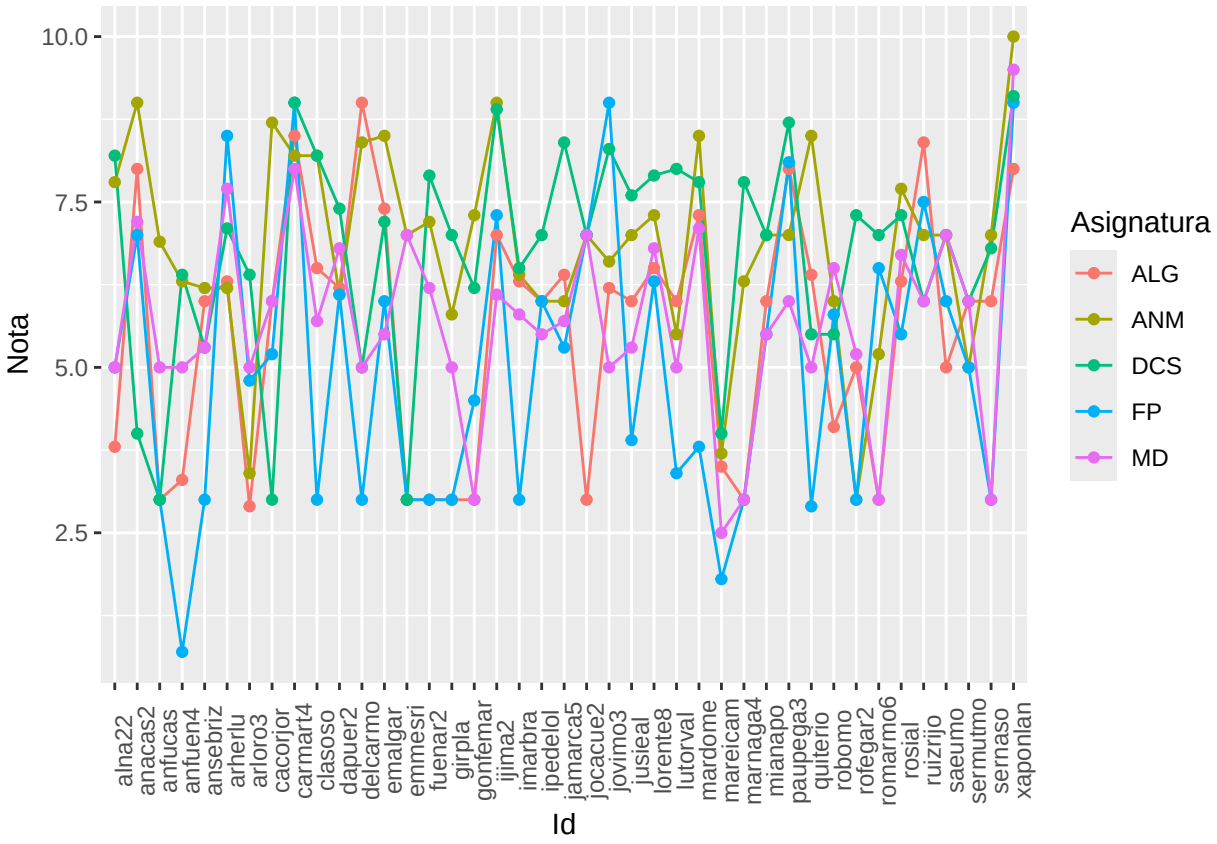


Figura 5: Representacion de alumnos con asignaturas completas

12. Sin hacer ninguna modificación en el código utiliza el fichero correspondiente al año 2025 y compara los resultados con los del curso 2022

No se han obtenido los mismos resultados. Habían valores/errores que no estaban considerados en el fichero 2022.

- Cantidad de estudiantes
- Valores faltantes
- Control sobre los valores de entrada (M, M , Masculino)

A pesar de que los valores son notablemente diferentes, se puede apreciar la correlacion entre la mediana y el percentil 75.

13. Responde a las siguientes preguntas, utilizando los datos del curso 2025. Ten en cuenta que deberás hacer correcciones en los datos en aquellos casos en los que los valores introducidos contengan errores. Por ejemplo si en las instrucción se indica que para el sexo de nacimiento los valores posibles son M y F, cualquier otro valor deberá ser corregido. Si la corrección no es posible sustituye por NA. Debes justificar todas las cuestiones con los datos disponibles.

Nota: Se han hecho los respectivos cambios para continuar con los siguientes apartados.

- ¿Qué edad, expresada en años (con decimales) tiene el alumno más joven a fecha 01/02/2025.

En principio, el formato de fecha debe ser dd-mm-yyy. Sin embargo, sabemos que se pueden presentar otros formatos, debido al error humano.

```
# EXPRESION REGULAR PARA VER LOS FORMATOS DE FECHA
# ^ inicio de la cadena
# \\d <- Será un digito
# \\d{n} <- n digitos
# $ fin de la cadena

# Quitar los espacios en blanco en la columna
datos <- datos %>%
            mutate(Fecha = str_trim(Fecha))

datos %>%
  # Ver que fechas siguen el patron '^\\d{2}-\\d{2}-\\d{4}$' (dd-mm-yyyy)
  filter(!if_any(.cols = everything(),
                 .fns = ~ str_detect(string = ., pattern = '^\\d{2}-\\d{2}-\\d{4}$')))
```

Fecha	Id	Age	Sex	Wr.Hnd	NW.Hnd	Fold	Pulse	Clap	Exer	Smoke	Height	A
20-7-2006	anfuen4	18	M	21	Derecha	Derecho	65	Der s Izq	8	Si	179	
23-7-2006	mareicam	18	M	17	Derecha	Derecha	62	Der s Izq		No	170	
18/09/2006	fuenar2	18	F	9,5	Derecha	Derecha	65	Izq s Der	3	No	169	
20-6-2006	ipedelol	18	M	16	Derecha	Derecho	65	Der s izq	10	No	173	
16-3-2006	ijima2	18	M	20	Derecha	Derecho	68	Izq s Der	No	No	172	
16-8-2006	jamarca5	18	M	20	Derecha	Izquierdo	65	Der s Izq	3	No	1,72	
21/12/2006	emmesri	18	M	16	Derecha	Izquierdo	65	Der s izq	si	No	160	
12-7-2005	sermutmo	19	M	20	Izquierda	Izquierdo	65	Izq s Der	Si	Si	180	

```r
  # Hay 8 registros con formato dd-m-yyyy o dd/mm/yyyy
```

Vamos a separar la columna fecha, para obtener los valores individuales de dia, mes y año, y luego los uniremos nuevamente en un formato estandarizado para convertir a la columna fecha en su formato correcto.

```r
datos <- datos %>%
            # Separamos las columnas
            separate(col = Fecha, into = c('dia', 'mes', 'anio'), sep = '[/-]') %>%

            # Convertimos a enteros para comprobar que los valores no estén fuera de rango
            mutate(across(.cols = dia:anio,
                          .fns = ~ as.integer(.)))

# Comprobamos
datos %>%
  select(dia:anio) %>%
  filter(dia > 31 | mes > 12 | anio > 2010)
```

dia	mes	anio

```r
  # De haber datos que cumplan esto, habría que tratarlos

# Unir los datos para convertir a fecha en el formato adecuado
datos <- datos %>%
            mutate(across(.cols = dia:anio,
                          .fns = ~ as.character(.))) %>%
            unite(col = Fecha, c(dia:anio), sep = '-') %>%
            # Para formato dd-mm-yyyy
            mutate(Fecha = as.Date(Fecha, format = '%d-%m-%Y'))
```

Ahora hacemos las operaciones correspondientes para la edad del alumno más joven a fecha 01/02/2025.

```r
# Asignamos la fecha para comparar
fecha <- '01/02/2025'
fecha_filtro <- as.Date(fecha, format = '%d/%m/%Y') # Formato en que se recibe
fecha_filtro <- as.Date(fecha_filtro, format = '%d-%m-%Y') # Formato en que se compara

# Operamos
datos %>%
  select(Fecha, Id) %>%
  # interval(fechainicio, fechafin), days(1), weeks(1), months(1)...
  mutate(Edad = round(interval(Fecha, fecha_filtro) / years(1), 2)) %>%
  arrange(Edad) %>%
  head(., 1)
```

Fecha	Id	Edad
2006-12-31	delcarmo	18.09

El estudiante más joven es ‘delcarmo’, con 18.09 años de edad.